

(19)日本国特許庁(JP)

(12)公開特許公報(A)

(11)特許出願公開番号

特開平5-298225

(43)公開日 平成5年(1993)11月12日

(51)Int.Cl.<sup>5</sup>

G 0 6 F 13/10

識別記号

3 1 0 B 8133-5B

庁内整理番号

F I

技術表示箇所

審査請求 未請求 請求項の数4(全 24 頁)

(21)出願番号 特願平4-99327

(22)出願日 平成4年(1992)4月20日

(71)出願人 000001007

キャノン株式会社

東京都大田区下丸子3丁目30番2号

(72)発明者 長島 正明

東京都大田区下丸子3丁目30番2号 キャ  
ノン株式会社内

(72)発明者 伊達 厚

東京都大田区下丸子3丁目30番2号 キャ  
ノン株式会社内

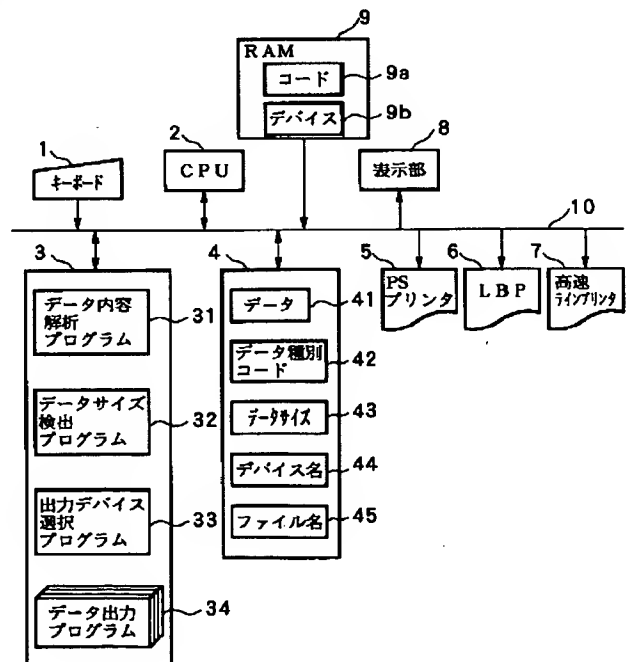
(74)代理人 弁理士 大塚 康徳 (外1名)

(54)【発明の名称】 データ処理方法及び該方法を用いる装置

(57)【要約】

【目的】 入力されたデータの種類、サイズ或いは属性の少なくとも1つに基づいて出力するデバイスを自動的に判別してそのデータに応じた出力デバイスにデータを出力できる。本発明はまた、入力されたデータのサイズ或いは属性の少なくとも1つに基づいて、そのデータを記憶するデバイスを自動的に判別してそのデータに応じた記憶デバイスにデータを出力できるデータ処理方法及び該方法を用いた装置を提供する。

【構成】 データ入力手段によりデータを入力し、その入力されたデータの種類、サイズ及び属性のうち少なくとも1つを判別する。この判別された判別結果により複数の出力デバイスの内いずれのデバイスを選択するかを決定し、その選択された出力デバイスにデータを出力する。



**【特許請求の範囲】**

**【請求項 1】** 複数の出力デバイスを接続するデータ処理装置であって、

データを入力するためのデータ入力手段と、

前記データ入力手段により入力されたデータの種別、サイズ及び属性のうち少なくとも 1 つを判別するためのデータ判別手段と、

前記データ判別手段の判別結果により前記複数の出力デバイスの内いずれのデバイスを選択するかを決定するデバイス選択手段と、

前記デバイス選択手段により選択された出力デバイスに前記データを出力するためのデータ出力手段と、  
を有することを特徴とするデータ処理装置。

**【請求項 2】** 複数の記憶デバイスを接続するデータ処理装置であって、

データを入力するためのデータ入力手段と、

前記データ入力手段により入力されたデータの種別、サイズ及び属性のうち少なくとも 1 つを判別するためのデータ判別手段と、

前記データ判別手段の判別結果により前記複数の記憶デバイスの内いずれの記憶デバイスを選択するかを決定するデバイス選択手段と、

前記データに対する前記選択手段により選択された前記記憶デバイスの記憶場所と前記データを記憶するための名称とを生成する生成手段と、

前記デバイス選択手段により選択された記憶デバイスに前記生成手段により生成された情報とともに前記データを記憶すべく出力するデータ出力手段と、  
を有することを特徴とするデータ処理装置。

**【請求項 3】** 複数の出力デバイスを接続して前記出力デバイスにデータを出力する方法であって、

データを入力する工程と、

その入力されたデータの種別、サイズ及び属性のうち少なくとも 1 つを判別する工程と、

その判別結果により前記複数の出力デバイスの内いずれのデバイスを選択する工程と、

その選択された出力デバイスに前記データを出力する工程と、

を有することを特徴とするデータ処理方法。

**【請求項 4】** 複数の記憶デバイスを接続して前記複数の記憶デバイスにデータを出力するデータ処理方法であって、

データを入力する工程と、

その入力されたデータの種別、サイズ及び属性のうち少なくとも 1 つを判別する工程と、

その判別結果により前記複数の記憶デバイスの内いずれの記憶デバイスを選択する工程と、

その選択された記憶デバイスの記憶場所と前記データを記憶するための名称とを生成する工程と、

その選択された記憶デバイスに、生成された情報とともに

前記データを記憶すべく出力する工程と、  
を有することを特徴とするデータ処理方法。

**【発明の詳細な説明】****【0001】**

**【産業上の利用分野】** 本発明は、複数のデバイスを接続して、そのデバイスにデータを出力するデータ処理方法及び該方法を用いる装置に関する。

**【0002】**

**【従来の技術】** 外部記憶装置やプリンタ等の出力デバイスの種類が増加するにつれて、これら多くの出力デバイスを接続して、出力データの大きさや種類によって、出力先のデバイスを変えて出力できる端末やシステムに対する要求が高まってきている。従来、このような複数の出力デバイスを接続できるシステムで出力デバイスを指定する場合には、ユーザはまずデータの種別をみて、目的とする出力デバイスに出力できるかどうかをチェックし、出力できるデータの種別であれば次にそのデータのサイズを調べ、その出力デバイスのデータサイズに納まるかどうかをチェックする。そして、そのデータをその出力デバイスに出力できると判断すると、その出力デバイス名を直接入力して指定するか、或いはその出力デバイスに出力するようなプログラムを起動する等して、直接もしくは間接的に出力デバイスを指定していた。

**【0003】**

**【発明が解決しようとする課題】** しかし上述した従来例では、データを出力デバイスに出力する場合、上述したように、一々その出力デバイスにデータ出力が可能かどうかをチェックした後、出力デバイスを指定する必要があった。データを出力デバイスに出力する場合、一般的にユーザはデータの種別や大きさ等のチェックを行わず、例えばプリンタや端末或いは外部記憶装置等に出力することが多い。そのため、例えば、プリンタや端末に不適切なデータを送ってしまったり出力デバイスをハングアップさせたり、あるいは出力デバイスの設定を狂わせてしまう等の虞があった。

**【0004】** また、データを外部記憶装置に保存する場合、自機により制御されているハードディスク（自分のマシンからしかアクセスできないものとする）と、サーバマシン（他の機器からもアクセスできる）下のハードディスクがそれぞれ接続されている場合、その出力デバイス（ハードディスク）の指定を行わないと、極秘データや作成中のプログラム等の私用データと、サーバマシンにより管理されるべき共用データとがそれぞれのハードディスクに誤って記憶されて混在し、データ管理が煩雑になってしまう虞もある。

**【0005】** 本発明は上記従来例に鑑みてなされたもので、出力しようとするデータの種別、サイズ或いは属性の少なくとも 1 つに基づいて出力するデバイスを自動的に判別してそのデータに適した出力デバイスにデータを出力できるデータ処理方法及び該方法を用いた装置を提

供することを目的とする。

【0006】本発明はまた、保存しようとするデータの種類、サイズ或いは属性の少なくとも1つに基づいて、そのデータを記憶するデバイスを自動的に判別してそのデータに適した記憶デバイスにデータを出力できるデータ処理方法及び該方法を用いた装置を提供することを目的とする。

【0007】

【課題を解決するための手段】上記目的を達成するために本発明のデータ処理装置は以下の様な構成を備える。即ち、複数の出力デバイスを接続するデータ処理装置であって、データを入力するためのデータ入力手段と、前記データ入力手段により入力されたデータの種類、サイズ及び属性のうち少なくとも1つを判別するためのデータ判別手段と、前記データ判別手段の判別結果により前記複数の出力デバイスの内いずれのデバイスを選択するかを決定するデバイス選択手段と、前記デバイス選択手段により選択された出力デバイスに前記データを出力するためのデータ出力手段とを有する。

【0008】上記目的を達成するために本発明のデータ処理方法は以下の様な工程を備える。即ち、複数の出力デバイスを接続して前記出力デバイスにデータを出力する方法であって、データを入力する工程と、その入力されたデータの種類、サイズ及び属性のうち少なくとも1つを判別する工程と、その判別結果により前記複数の出力デバイスの内いずれのデバイスを選択する工程と、その選択された出力デバイスに前記データを出力する工程とを有する。

【0009】

【作用】以上の構成において、データ入力手段によりデータを入力し、その入力されたデータの種類、サイズ及び属性のうち少なくとも1つを判別する。この判別された判別結果により複数の出力デバイスの内いずれのデバイスを選択するかを決定し、その選択された出力デバイスにデータを出力するように動作する。

【0010】

【実施例】以下、添付図面を参照して本発明の好適な実施例を詳細に説明する。

【0011】まず、図1～図4を参照して本発明の第1の実施例について説明する。この第1の実施例では、データの種類及びサイズをみて、異なる出力デバイスに出力する例を示し、この実施例では、データの種類として、ポストスクリプトデータ、画像データ、テキスト（リスト）データを考える。このポストスクリプトデータとは、Postscript Language Reference Manual (Adobe Systems Inc. 著)に基づいて記述されたページ記述言語データであり、ここでは、上著で推奨しているように“%!”で始まっているものとする。

【0012】また、画像データは、TIFF (Tag Image File Format) (“Windows の普及で注目される画像フォ

ーマット、TIFF”，日経CG, 1989.10参照)に基づいて記述された画像データで、CPUによって値が異なるが、初めにマジックコード（例えば、インテル系で16進数で“4949”、モトローラ系で16進数で“4d4d”）がある。ここでは、CPUがインテル系であるとし、16進数で“4949”、即ちアスキーコードに基づいて変換した文字列“II”で始まるものとする。また、テキストデータは、印字可能な文字列からなるデータで、ポストスクリプトデータ、TIFFデータでないもの、即ち、文字列“%!”，“II”で始まらないものとする。

【0013】このように本実施例では、上記3つの種類のデータのみを扱うことにする。また出力デバイスとして、ポストスクリプトデータのみを扱うポストスクリプトプリンタ、テキストデータと画像データをプリントアウトできるレーザービームプリンタ (LBP)、テキストデータのみしか扱えないが非常に高速にプリントアウトできる高速ラインプリンタを考える。

【0014】図1は、第1の実施例の情報処理システムの概略構成を示すブロック図である。

【0015】図において、1はキーボードで、オペレータにより操作され、このシステムにコマンドや或いはテキストデータを入力することができる。2はこの情報処理システムにおいて各種処理を行うためのCPU、3は、CPU2により実行される各種処理の動作手順が記述されているプログラムを保管するための第1の記憶装置、4は入力されたデータ等を一時的に保管するための第2の記憶装置である。5はポストスクリプトデータのみを処理して印刷できるポストスクリプト・プリンタ、6はテキストデータと画像データをプリントアウトできるLBP、7は高速ラインプリンタである。8は表示部で、オペレータへのメッセージやキーボード1等より入力されたデータ等を表示する。9はCPU2のワークエリアとして使用されるRAMで、後述するコードデータを記憶するための変数エリア（コード9a）や出力デバイス名を記憶するデバイス9b等を備えている。これらがネットワーク10に接続されている。尚、第1の記憶装置3と第2の記憶装置4は、ここでは別の構成として示しているが、同一のものであってもさしつかえないことはもちろんである。

【0016】次に、第1の記憶装置3に記憶されている各種プログラムについて説明する。31はデータ内容解析プログラムで、入力されたデータ41の内容を見て、データ41がバイナリデータ、即ち画像データか、ポストスクリプトデータか、それともテキストデータかを判定し、その結果をデータ種別コード42に記憶する。32はデータサイズ検出プログラムで、各種データのサイズを検出し、その検出結果をデータサイズ43に記憶する。33は出力デバイス選択プログラムで、第2の記憶装置4に記憶されたデータ種別コード42とデータサイ

ズ43との内容を参照して、出力データを記憶しているデータ41の内容を出力すべき出力デバイスを選択する。34は各出力デバイスごとに設けられたデータ出力プログラム群で、出力デバイス選択プログラムによって選択された出力デバイスの種類によっては、出力データをそのまま何の加工もせずを選択された出力デバイスに出力するプログラムもあれば、選択された出力デバイス特有の制御コマンド等を出力データに付加する等の処理を行って出力データを加工するプログラムなども含んでいる。

【0017】次に、第2の記憶装置4の各種データエリアについて説明する。41はデータファイルで、キーボード1より入力されたテキストデータ（ポストスクリプトデータを含む）或いはネットワーク8を通じて本システムに入力されたテキスト（ポストスクリプトデータを含む）もしくはバイナリデータ（画像データ等）等の出力データを記憶している。42はデータ種別コードで、データ内容解析プログラム31によって判別された、出力データの種別を示す種別コード、例えばデータ41がテキストデータなら種別コード“0”，画像データなら種別コード“1”，ポストスクリプトデータなら種別コード“2”が割当てられて記憶されている。尚、この種別コードは、出力データの種類によって割当てられるコードで、本実施例のように数字である必要はなく、例えば文字列であってもよい。43はデータサイズで、データサイズ検出プログラム32によって検出されたデータ41のデータサイズが記憶される。44はデバイス名で、出力デバイス選択プログラム33によって選定された出力デバイスの名称（或いはそれを表す記号や数字等）記憶される。45はファイル名で、キーボード1より出力が指示されたデータファイルの名称が記憶されている。

【0018】次に、図2のフローチャートを参照して、第1の実施例の情報処理システムにおける処理の流れについて説明する。図中、S201からS206は各処理ステップを表わしている。

【0019】まずステップS201で、キーボード1よりデータ41に記憶されているデータの内、出力したいデータのファイル名が入力されると、そのファイル名をファイル名45として第2の記憶装置4に記憶する。出力データを決定する方法は、これに限定されるものでなく、例えばファイルに番号が対応付けられていれば、その番号をキーボード1から入力してファイルを指示しても良く、或いはキーボード1の代わりにマウスのようなポインティングデバイスを利用して出力データのファイルを指定してもよい。

【0020】次にステップS202に進み、データ内容解析プログラム31を用いて、ファイル名45によって指示されたデータファイルのデータ41がポストスクリプトデータであるか、テキストデータであるか、それと

もバイナリデータであるかを判定し、その判定結果をデータ種別コード42として第2の記憶装置4に記憶する。このステップS202の詳細な動作については図3のフローチャートを参照して後述する。

【0021】次にステップS203に進み、データサイズ検出プログラム32を用いて、指定されたデータ41のサイズ、例えばバイト数を求め、これをデータサイズとして第2の記憶装置4のデータサイズ43に記憶する。これは、例えばUNIXシステムにおけるwc（ワードカウンタ）コマンドによりファイルのサイズを検出することが実現されているので、これを利用してよいし、あるいはアプリケーションプログラム等によりワード数をカウントしてもよい。次にステップS204に進み、出力デバイス選択プログラム33を用いて、データ種別コード42及びデータサイズ43に基づいて出力デバイス（プリンタ）を決定し、デバイス名44として第2の記憶装置に保管する。このステップS204の詳細は後述する。次にステップS205に進み、データ出力プログラム群34の中から、ステップS204で決定された出力デバイス名44に対応するデータ出力プログラムを選択してデータ41をその出力プログラムに入力した後、対応する出力デバイスに転送することにより、データを出力する。この時、どのプリンタに出力したかを、メッセージとして表示部8に表示してオペレータに知らせることにより、より効果的になる。

【0022】尚、上述の説明では、ステップS204とステップS205の処理とをそれぞれ別の処理として示したが本発明はこれに限定されるものでなく、出力デバイス選択プログラム33によって出力デバイスを選択し、それに対応する出力プログラムを選択するようになれば、デバイス名44を記憶する必要がなく、ステップS204とステップS205の処理を分ける必要もなくなる。また、ステップS202でテキストデータでないと判断されたなら、ステップS203を省いてもよい。

【0023】次に、図3のフローチャートを参照して、ステップS202のデータ内容の判別処理について説明する。このデータ内容の判別方法は、各データの形がどのようなものであるかによってその判定アルゴリズムが異なるが、ここでは上述したように、ポストスクリプトデータ、TIFFデータ、テキストデータのいずれであるかの判別を行なうものとする。図3において、S301からS309は各処理ステップを表わしている。

【0024】まずステップS301で、データ41の最初の2バイトのデータを取り込む。但し、1バイト目がEOF（End Of File）の時は、2バイトのデータを取り込めないで、1バイトだけ取り込むことになる。そしてステップS302で、この取り込んだデータに“EOF”が含まれるかどうかを判定し、“EOF”コードが含まれる時はステップS305に進み、1バイト目が“EOF”かどうかをチェックをする。このステップS

305で、1バイト目が“EOF”の時はステップS306に進んで、その旨をメッセージを表示部8に出力して処理を終了する。一方、ステップS305で1バイト目が“EOF”でない時、即ち2バイト目が“EOF”の時は、データ41はテキストデータであるとしてテキストデータの種別コードをRAM9のコード9aに記憶する。このコード9aに記憶されたコードデータは、ステップS310で第2の記憶装置4のデータ種別コード42に記憶される。そして、メインルーチンにリターンする。

【0025】一方、ステップS302で“EOF”を含まない時はステップS303に進み、2バイトデータが16進数の“4949”、即ちASCIIコードに対応した文字列“II”かどうかを判定する。ステップS303で“II”であると判定されるとステップS307に進み、データ41はTIFFデータ、即ち画像データであるとして、画像データであることを示す種別コードをコード9aに格納してステップS310に進む。

【0026】また、ステップS303で文字列“II”でない時はステップS303に進み、2バイトデータが文字列“%!”かどうかを判別する。文字列“%!”の時はステップS308に進み、データ41はポストスクリプトデータであるとして、ポストスクリプトデータであることを示す種別コードをコード9aに記憶してステップS310に進む。また一方、ステップS304で“%!”でない時はステップS309に進み、前述したようにコード9aにテキストデータであることを示す種別コードを記憶する。こうして図2におけるステップS202の処理の説明を終える。

【0027】次に図4のフローチャートを参照して、図2のステップS204の出力デバイスの選択処理について説明する。図中、S401からS409は各ステップを表わしている。

【0028】まずステップS401で、第2の記憶装置4に記憶されているデータ種別コード42を読取り、ステップS402でそのデータ種別コードが、ポストスクリプトデータのコードかどうかを判定する。そうであればステップS406に進み、ポストスクリプトプリンタを選択し、そのプリンタ名をRAM9の変数デバイス9bに記憶する。そして、ステップS409で、このデバイス9bのデータを第2の記憶装置にデバイス名44として登録する。

【0029】一方、ステップS402で、データ種別コード42の内容がポストスクリプトデータでないならステップS403に進み、画像データかどうかを判定する。画像データのコードであればステップS407に進んでLBPを選択し、そのプリンタ名をデバイス9bに記憶し、その後ステップS409に進む。

【0030】一方、ステップS403で、データ種別コード42の内容が画像データでないならステップS40

4に進み、第2の記憶装置4に記憶されているデータサイズ43を入力する。次にステップS405で、このデータサイズ43の値が予め決められている閾値より大きいかどうかを判定し、大きい時はステップS408に進んで高速ラインプリンタを選択し、そのプリンタ名をデバイス9bに格納してステップS409に進む。そうでない時はステップS407に進み、LBPを選択し、そのプリンタ名をデバイス9bに記憶する。

【0031】尚、ステップS405において判定に使用される閾値は、予めシステムで決定されていても良く、或いはユーザが任意に指定できるようにしてもよい。これで図2におけるステップS204の処理の説明を終了する。

【0032】このように第1の実施例によれば、出力するデータの種類やサイズによって、最適な出力デバイスを自動的に選択して出力できる効果がある。

【0033】次に図5～図7を参照して、本発明の第2の実施例について説明する。この第2の実施例では、データの属性、特にアクセス権に関する属性によってデータの出力先のデバイスを切り替える例を示している。この第2の実施例では、データの属性として、自分だけがアクセスできるデータと、他人もアクセス可能であるデータの両方が存在する場合で考える。もちろん、例えばUNIXのように、アクセス対象のユーザを自分、所属グループ、他人としたり、あるいはアクセス権の種類をリード、ライト、実行というように細分してもかまわない。また、出力デバイスとして、自分のマシンからしかアクセスできない私用のハードディスクと、どのマシンからもアクセスできる共用のハードディスクを考える。そして、データを作成し終えて、それをセーブする時に、そのデータのアクセス権に応じてセーブする先のハードディスクを選択する。

【0034】図5は、本発明の第2の実施例の情報処理システムの概略構成を示すブロック図で、図1と共通する部分は同じ番号を付し、それらの説明を省略する。

【0035】図5において、15は各処理を行なうための第1のCPU、16は第1のCPU15から要求があった時、特定のデータをハードディスク18に保存する処理を行なう第2のCPUである。17は第1のCPU15からのみアクセスできるハードディスク、即ち、第1のCPU15の利用者のみしかアクセスできない個人用ハードディスク、18はどのCPUからアクセス可能な共用ハードディスク、19は第2のCPU16が共用ハードディスク18にセーブするプログラム191を保管するための第3の記憶装置である。

【0036】次に、第1の記憶装置3aに記憶された制御プログラムについて説明する。35はデータ属性指定プログラムで、ユーザにセーブすべきデータの属性を入力させるためのプログラムで、その入力されたデータ属性は第2の記憶装置4aのデータ属性46に記憶され

る。36は出力デバイス選択プログラムで、データ属性46の内容をみて、その結果からデータ41を出力すべき出力デバイスを選択する。37はデータ出力プログラムで、個人用ハードディスク17にデータ41をセーブするための制御を行い、データ転送プログラム38は第2のCPU16に対してデータセーブ要求とデータ41を転送する。39はデータ編集プログラムで、データ41を作成するための編集処理等を行う。

【0037】次に第2の記憶装置4aの各データについて説明する。41はデータ編集プログラム39により作成されたデータ、44は図1と同様に、出力デバイス選択プログラム36によって選定されたデバイス名である。45は図1と同様に、キーボード1より入力されたファイル名である。46はデータ属性指定プログラム35によって指定されたデータ41のデータ属性、47は過去にハードディスク17、18にセーブしてきたデータの属性情報が格納されている属性テーブルである。この属性テーブル47には、データファイル名とその属性情報が格納されており、このデータファイル名を手がかりに、その属性を取り出せるようになっている。また、第3の記憶装置19において、191は第1のCPU15から転送されたデータを共用ハードディスク18にセーブするための第2のデータ出力プログラムである。

【0038】次に、この第2の実施例の情報処理システムにおける処理の流れを図6のフローチャートを用いて説明する。尚、この処理は、データ編集プログラム39を用いてデータ41を第2の記憶装置4aに作成し、このデータ41をハードディスク17或いは18にセーブする際に行なう処理を示している。図中、S601からS607は各処理ステップを表わしている。

【0039】まずステップS601で、出力したいデータをハードディスクにセーブする時のファイル名をキーボード1より入力し、この入力されたファイル名を第2の記憶装置4aにファイル名45として格納する。次にステップS602に進み、データ属性指定プログラム35により、データのアクセス権に関する属性を指定して第2の記憶装置4aにデータ属性46として登録するとともに、属性テーブル47にも、ステップS601で指示されたファイル名と属性情報を付加する。但し、すでに属性テーブル47内にそのデータに関する属性情報が格納されている場合は、その情報を更新することになる。この属性テーブル47は、この図6に示す処理の終了後も存在すべき情報であるので、第2の記憶装置4aが一時的な記憶しかできない場合は、処理の終了後に永続的な記憶装置、例えば、第1の記憶装置3aに更新された内容を保存しておき、処理に際して第2の記憶装置4aに読み出すようにすればよい。

【0040】次にステップS603に進み、データ属性46の内容により、出力デバイス選択プログラム36を用いて出力デバイス（ハードディスク）を決定し、デバ

イス名44として第2の記憶装置4aに記憶する。次にステップS604に進み、第2の記憶装置4aのデバイス名44を入力し、データ属性46の内容がCPU15だけしかアクセスを許さないということであれば個人用ハードディスク17を選択し、他のCPUにもアクセスを許すということであれば、共用ハードディスク18を選択する。尚、ここでは、ステップS603とステップS604の処理をそれぞれ独立に示したが、本発明はこれに限定されるものでなく、出力デバイス選択プログラム36において、デバイス名44を出力せずに、ステップS603の処理の内部でステップS604の処理、即ち、個人用ハードディスク17を指しているかどうかの判定を行なってもよい。

【0041】こうしてステップS604で、個人用ハードディスク17に記憶する時はステップS605に進み、データ41を個人用ハードディスク17に出力してセーブし、処理を終了する。一方、ステップS604で個人用ハードディスク17にセーブしない時はステップS606に進み、第2のCPU16にファイルセーブの要求を出し、ステップS607でデータ41の内容をCPU16に転送する。そして、第2のCPU16に処理を委ねて終了する。

【0042】次に、これに対応して動作する第2のCPU16の処理の流れについて、図7のフローチャートを用いて説明する。

【0043】ステップS701で、第2のCPU16は第1のCPU15からファイルセーブ要求がきたかどうかをチェックする。ファイルセーブ要求を入力するとステップS702に進み、その後に転送されてくるデータを共用ハードディスク18に出力してセーブする。その後、ここで終了してもよいし、あるいはフローチャートにあるように再びステップS701に戻って、第1のCPU15からの次のファイルセーブ要求の待ち状態に入ってもよい。また、第1のCPU15にファイルセーブ処理が終了した等のメッセージを送るようにしてもよい。

【0044】次に、こうしてハードディスクに記憶されたデータファイルより所望のデータを讀出して第2の記憶装置4a等にロードする時は、属性テーブル47を参照することにより、そのファイルの属性データから、そのデータファイルが個人用ハードディスク17に記憶されているか、或いは共用ハードディスク18に記憶されているかを知ることができる。そこで、個人用ハードディスク17に記憶されている時は、その個人用ハードディスク17より読み出せば良く、また共用ハードディスク18に記憶されている時は、第2のCPU16に対してロード要求を出力して所望のデータファイルを讀出するように指示し、第2のCPU16により讀出されたデータをネットワーク10を介して取り込めば良い。

【0045】以上述べたように第2の実施例によれば、

個人用のデータを間違えて共用ハードディスクに、或いは逆に共用データを間違えて個人用ハードディスクにセーブすることがなくなる。これにより個人用のデータを間違いなく完全に隔離して記憶することができる。尚、上記実施例はデータを保存する場合について述べたが、第1の実施例のようにデータを印刷する場合にも、上記のような属性情報に基づいて、個人用のプリンタに出力するのか共用のプリンタに出力するのかを判別させることもできる。次に図8～図10を参照して、本発明の第3の実施例の情報処理システムについて説明する。尚、この図8では前述の図と共通する部分は同じ番号で示し、それらの説明を省略する。

【0046】図8において、21、22はデータファイルとしてデータをセーブするためのハードディスクで、21は比較的小容量の磁気ハードディスク、23は大容量の光磁気ハードディスクである。第1のメモリ（記憶装置）3bは、CPU2の動作を規定する各種制御プログラムを記憶しており、第2のメモリ（記憶装置）4bは、各処理において生成される情報を一次的に保存して、CPU2のワークエリアとして使用されている。

【0047】次に、第1のメモリ3bに記憶されている各種制御プログラムについて説明する。

【0048】51はデータ編集プログラムで、第2のメモリ4bに記憶されるデータ61を作成する。52はデータサイズ検出プログラムで、データ編集プログラム51により作成されたデータ61のデータサイズを検出して、第2のメモリ4bのデータサイズ63に記憶する。53は出力デバイス選択プログラムで、データサイズ63に記憶されたデータ61のデータサイズに応じて、そのデータ61を出力する出力デバイスを、ハードディスク21にするか、光磁気ディスク22にするかを選択する。54は保管場所特定プログラムで、データ61をハードディスク21或いは22にセーブする時、その保管場所（ディレクトリ）を特定する。55はデータセーブプログラムで、データ61を出力デバイス選択プログラム53により選択されたハードディスクの、保管場所特定プログラム54により特定されたディレクトリにセーブする。56はデータロードプログラムで、ハードディスク21或いは光磁気ディスク22よりのデータの読出しを制御している。

【0049】次に、第2のメモリ4bの構成を説明する。

【0050】データ61は、前述したようにデータ編集プログラム51により作成されたデータである。62はキーボード1より入力されたファイル名で、このファイル名とともに、データ61がハードディスク21或いは光磁気ディスク22に記憶される。63はデータサイズで、データサイズ検出プログラム52により検出されたデータ61のサイズ（例えばバイト数）が記憶される。64はディレクトリ名で、保管場所特定プログラム54

により特定されたディレクトリの名称が記憶される。

【0051】次に図9のフローチャートを参照して、この第3の実施例の情報処理システムにおけるデータセーブ動作を説明する。尚、この処理の前には、データ編集プログラム51によりデータが作成されて、第2のメモリ4bのデータ61に記憶されており、このデータ61をディスク21或いは22に記憶するものとする。

【0052】まずステップS801で、データ61をセーブする時のファイル名をキーボード1より入力し、その入力されたファイル名を第2のメモリ4bのファイル名62に記憶する。次にステップS802に進み、データサイズ検出プログラム52により、データ61のデータサイズ（例えばバイト数）を検出し、その検出結果をデータサイズ63に記憶するとともに、RAM9のサイズ90にもカレントディレクトリとして記憶する。そしてステップS803に進み、このデータサイズ63の値と閾値Lとを比較し、その判定結果に応じて、どのディスクに出力するかを決定する。

【0053】ステップS803で閾値Lよりも小さい時はステップS804に進み、カレントディレクトリにデータ61をファイル名62で指示されたファイル名で記憶する。ここで、この閾値Lは、予め定められた値であっても良く、或いはオペレータがキーボード1より入力した任意の値であっても良い。また、このカレントディレクトリは、RAM9に設けられていても良く、或いはハードディスク21或いは比較磁気ディスク22に設けられていても良い。

【0054】一方、ステップS803で、データサイズ63の値が閾値Lよりも大きい時はステップS805に進み、保管場所を特定する保管場所特定プログラム54により保管場所（光磁気ディスク22が割り当てられているディレクトリ）を決定し、その保管場所にデータ61を、ファイル名52に記憶されている名称でセーブする。そしてステップS807で、ディレクトリ名64をカレントディレクトリにセーブして処理を終了する。但し、ここで過去に光磁気ディスク22のほうにセーブしたファイルが存在する時は、カレントディレクトリには既にディレクトリ名が作成されているので、ステップS807でディレクトリ名をセーブする必要はない。

【0055】尚、ステップS805における保管場所の決定方法は、予め保管場所を固定しておいても良く、或いは光磁気ディスク22に保管する時の保管場所を記述したテーブルをユーザが作成しておき、保管場所特定プログラム54がそのテーブルを参照して決定するようにしても良い。

【0056】こうしてハードディスク21或いは光磁気ディスク22に記憶されたデータファイルより、所望のデータを読出して第2のメモリ4bにロードする時の処理手順を図10のフローチャートを参照して説明する。尚、この処理はデータ61を作成したディレクトリ下で

の処理を示している。

【0057】まずステップS811で、キーボード1よりロードしたいデータのファイル名が入力されるとステップS812に進み、その入力されたファイル名がカレントディレクトリに存在するかどうかをみる。存在する時はステップS813に進み、そのディレクトリの対応するファイルよりデータをロードする。一方、そのファイル名が存在しない時はステップS814に進み、図9のステップS807でセーブしたディレクトリ名を読み込み、そのディレクトリの中の対応するファイルをロードして処理を終了する。

【0058】尚、この実施例では、大容量ディスクとして光磁気ディスクを例示しているが、本発明はこれに限定されるものでない。

【0059】このように第3の実施例によれば、データサイズにより、そのデータの保管場所、即ち、ハードディスクにセーブするか、或いは大容量の光磁気ディスクにセーブするかを自動的に選択してセーブするので、大容量のデータを容量の小さいハードディスクにセーブする等の間違いをなくすることができる。

【0060】図11は本発明の第4実施例の情報処理システムの概略構成を示すブロック図で、前述の図5と共通する部分は同じ番号で示している。この第4実施例では、データの属性、特にアクセス権に関する属性情報によって、出力先のデバイスを切り替える例を示している。尚、この実施例では、データの種類として、例えばCPU15だけがアクセスできるようにしたいデータと、他のCPU16もアクセスできるようにしたいデータとが存在するものとして考える。尚、もちろん例えばUNIXのように、アクセス対象のユーザを自分、所属グループ、他人としたり、或いはアクセス権の種類をリード、ライト、実行というように細分しても良い。また、出力デバイスとして、CPU15からしかアクセスできない私用のハードディスク17と、どのCPU（ワークステーション）からもアクセスできる共用のハードディスク18が接続されており、データをセーブする際に、そのデータのアクセス権に応じてどのハードディスクにセーブするかを選択できるものとする。そして、このセーブに際して、個人用ハードディスク17にセーブする時はカレントのディレクトリにセーブし、共用ディスク18にセーブする時はデータのあるホストマシン名、ディレクトリ名が示す保管場所にセーブするものとする。

【0061】第1のメモリ3cの構成を説明すると、501はデータ編集プログラムで、出力するためのデータを作成して第2メモリ4cのデータ601に記憶する。502はデータ属性指定プログラムで、作成されたデータ601の属性を指定し、この指定された属性データはデータ属性603に記憶される。503は出力デバイス選択プログラムで、データ属性603の内容より、デー

タ601を出力すべき出力デバイスを選択する。504はカレントホスト検索プログラムで、第1のCPU15を有するワークステーションのマシン名であるホスト名を検索する。505はカレントディレクトリを検索するためのカレントディレクトリ検索プログラム、506はデータセーブプログラムで、個人用ハードディスク17にデータをセーブする。507はデータ転送プログラムで、データセーブ要求と保管場所付きファイル名607とデータ601とを第2のCPU16に転送する。508は保管場所付きファイル名生成プログラムで、ホスト名605、ディレクトリ名606、ファイル名602から保管すべき場所を含むファイル名を作成する。509はデータロードプログラムで、データをハードディスク17からロードするためのプログラムである。

【0062】次に第2のメモリ4cのデータ構成について説明する。

【0063】データ601はデータ編集プログラム501により作成されたデータ、602はキーボード1より入力されたファイル名である。603はデータ属性指定プログラム502によって指定されたデータ601のデータ属性である。604は属性テーブルで、過去においてハードディスクにセーブしてきたデータの属性情報が記憶されている。605はカレントホスト検索プログラム504によって生成されるホスト名、606はカレントディレクトリ検索プログラムによって生成されるディレクトリ名である。607は保管場所ファイル名で、保管場所ファイル名生成プログラム509によって生成される。

【0064】次に、第3のメモリ3dの構成について説明する。

【0065】1001は保管場所付きファイル名解析プログラムで、保管場所付きファイル名をホスト名、ディレクトリ名、ファイル名に分解する。1002はディレクトリ生成プログラムで、指定された名前のディレクトリを作成する。1003はデータセーブプログラムで、第1のCPU15から転送されたデータを共用ハードディスク18にセーブする。1004はデータロードプログラムで、共用はハードディスク18よりデータをロードする。1005はデータ転送プログラムで、第1のCPU15にデータを転送する。

【0066】第4のメモリ4dの構成を説明すると、1101は第1のCPU15から転送されてきたデータを示し、1102は保管場所付きファイル名で、第1のCPU15より転送されて記憶されている。1103はディレクトリ名で、保管場所付ファイル名解析プログラム1001により解析されたホスト名及びディレクトリ名で、これら両者を合わせてディレクトリ名と呼ぶことにする。1104は保管場所付ファイル名解析プログラム1001により解析されて記憶されているファイル名である。



【0067】次に図5のフローチャートを参照して、この第4実施例の動作を説明する。この処理は、ファイルシステムがディレクトリ構造を持つシステムにおいて、あるディレクトリ下でデータ編集プログラム501を用いてデータ601を第2メモリに作成し、そのデータ601をハードディスク（17或いは18）にセーブする処理を示している。

【0068】まずステップS821で、キーボード1よりデータ601をセーブする時のファイル名が入力されると、そのファイル名を第2のメモリ4cにファイル名602として記憶する。次にステップS822において、データ属性指定プログラム502により、データ601のデータアクセス権に関する属性が指定されると、第2のメモリ4cにデータ属性603として記憶すると共に、属性テーブル604にもファイル名と属性情報を付加する。但し、既に属性テーブル604に、そのデータに関する属性情報がある時は、その情報を更新することになる。この属性テーブル604は、この実施例の終了後も存在すべき情報であるため、カレントディレクトリにセーブしておく。また、ここで、データ属性指定プログラム502は、セーブすべきデータ601の属性を、ユーザが入力するためのプログラムである。

【0069】次にステップS823に進み、そのデータ属性603の内容をチェックし、個人用のデータか、即ち、CPU15だけがアクセス権を有するデータかを調べ、そうであればステップS830に進み、個人用ハードディスク17の現在のディレクトリ、即ち、データ601を作成した場所のディレクトリに、データセーブプログラム506によりデータ601をセーブする。

【0070】一方、ステップS823で個人用データでない時、即ち、他人に対してもアクセス権の所有を許可することを示す属性の時はステップS824に進み、カレントホスト名検索プログラム504によって、第1のCPU15を有するホストマシンの名前を取り出し、その名前をホスト名605として第2のメモリ4cに記憶する。このホストマシン名を検索するプログラムとしては、例えばUNIXシステムのコマンド“hostname”がある。よって、このコマンドを用いても良く、或いはこのコマンドを参考にして作成したアプリケーションプログラムであっても良い。

【0071】次にステップS825に進み、カレントディレクトリ検索プログラム505によって、現在作業中のディレクトリを検索し、そのディレクトリ名をディレクトリ名606として第2のメモリ4cに記憶する。このカレントディレクトリを検索するプログラムとしては、例えばUNIXシステムにおける“pwd”コマンドがある。よって、このプログラムを利用しても良く、或いは新たに作成したプログラムであっても良い。次にステップS826において、ファイル名602、ホスト名605及びディレクトリ名606より、保管場所ファイ

ル名生成プログラム508によって、保管場所付ファイル名607を作成し、第2のメモリ4cに記憶する。この時、ホスト名605とディレクトリ名606はどちらか片方だけであっても良い。

【0072】次にステップS827に進み、第2のCPU16にファイルのセーブ要求を出力し、次に保管場所ファイル付ファイル名607、そしてデータ601を第2のCPU16に転送して第のCPU16に処理を委ねて処理を終了する。

【0073】次に図13のフローチャートを参照して、第2のCPU16によるデータセーブ処理について説明する。

【0074】まずステップS841で、第1のCPU15からファイルセーブ要求を受け取ったかどうかを調べ、受け取るとステップS842に進み、その次に転送されてくる保管場所付ファイル名607を第4のメモリ4dの保管場所付ファイル名1102に記憶する。そしてステップS843に進み、次に転送されてくるデータを受け取ってデータ1101に記憶する。次にステップS844に進み、保管場所ファイル名解析プログラム1001を用いて、保管場所付ファイル名1102をディレクトリ名とファイル名とに分解し、ディレクトリ名1103及びファイル名1104に記憶する。この時、ホスト名はディレクトリ名として扱う。即ち、ホスト名と同じ名前のディレクトリを共有ハードディスク18上に作成することになる。こうして得られたディレクトリ名1103が第2のCPU16側のシステムのファイル中に存在するかどうかをチェックする（ステップS845）。

【0075】ステップS845でディレクトリ名が存在する時はステップS847に進み、そうでない時はステップS846に進む。ステップS846では、ディレクトリ生成プログラム1002を用いて、そのディレクトリの生成を行う。このディレクトリ生成プログラムは、例えばUNIXシステムにおいて“mkdir”コマンドで実現されているので、これを利用するか、或いは新たにプログラムを作成しても良い。そしてステップS847において、データセーブプログラム1003を用いて、目的のディレクトリにデータ1101をセーブする。その後、ここで処理を終了しても良く、或いはフローチャートに示すように再びステップS1に戻り、第1のCPU15からのファイルセーブ要求の待ち状態に入っても良い。また、第1のCPU15に処理が終了したことを示すメッセージを送っても良い。

【0076】こうしてハードディスクに格納されたデータをロードする時の動作について以下に説明する。図14は第1のCPU15によるデータロード処理を示すフローチャートである。

【0077】第1のCPU15側のシステムでは、まずステップS851で、データ601を作成したディレク

トリ下において、まずロードすべきデータファイル名を入力し、ファイル名602に記憶する。そして、そのディレクトリ下にある属性テーブル604を読み込む。次にステップS853でそのデータ属性のチェックを行い、それが自システムのみがアクセス可能であることを示す属性であればステップS860に進み、データロードプログラム509を用いて、そのまま個人用ハードディスク17よりデータファイルをロードして処理を終了する。

【0078】一方ステップS853で、他のシステムに対してもアクセス権を与えることを示すデータ属性の時はステップS854に進み、カレントホスト検索プログラム504によりホスト名を、ステップS855でカレントディレクトリ検索プログラム505によりディレクトリ名を、ステップS856で保管場所付ファイル名生成プログラム508により保管場所付ファイル名を生成し、それぞれをホスト名605、ディレクトリ名606、保管場所付ファイル名607に格納する。そしてステップS857に進み、第2のCPU16にファイルのロード要求を出力し、次にステップS858で保管場所付ファイル名607を転送する。そしてステップS859で、第2のCPU16からのデータが届くのを待ち、受け取ると処理を終了する。

【0079】次に図15のフローチャートを参照して、第2のCPU16におけるロード処理を説明する。

【0080】まず第1のCPU15からロード要求が送られてきたかどうかをチェックし、ロード要求を入力するとステップS872に進み、その後送られてくる保管場所付ファイル名を、第4のメモリ4dの保管場所付ファイル名1102に格納する。そして次にステップS873に進み、保管場所付ファイル名解析プログラム1001を用いて、保管場所付ファイル名1102をディレクトリ名とファイル名とに分解し、それぞれをディレクトリ名1103とファイル名1104に格納する。次にステップS874に進み、データロードプログラム1004を用いてディレクトリ名1103が示すディレクトリ下のファイル名1104が示すファイルを共有ハードディスク18よりロードする。こうしてロードしたデータをデータ1101に格納し、ステップS875でデータ転送プログラム1005により第1のCPU15に転送する。その後、このロード処理を終了しても良く、或いはフローチャートに示すように再びステップS871に戻り、第1のCPU15からの要求待ちに移行しても良い。

【0081】以上説明したように第4の実施例によれば、個人用のデータと共有のデータとを間違えて異なるハードディスクに記憶することがなくなる。即ち、個人用のデータを完全に隔離することにより、他人に知らせないで済む。しかも各CPU側のシステムがディレクトリ構造を持つファイルシステムであっても対処できる効

果がある。また、第2～4の実施例では、データの属性またはサイズによって記憶先を判断する例を説明したが、第1の実施例のように、画像であるかテキストであるかといったデータの種類の基について判断するようにしてもよい。

【0082】尚、本発明は複数の機器から構成されるシステムに適用しても、1つの機器からなる装置に適用しても良い。また、本発明はシステム或は装置に、本発明を実施するプログラムを供給することによって達成される場合にも適用できることは言うまでもない。

【0083】

【発明の効果】以上説明したように本発明によれば、出力しようとするデータの種類、サイズ或いは属性の少なくとも1つに基づいて出力するデバイスを自動的に判別してそのデータに適した出力デバイスにデータを出力できる効果がある。

【0084】また他の発明によれば、保存しようとするデータの種類、サイズ或いは属性の少なくとも1つに基づいて、そのデータを記憶するデバイスを自動的に判別してそのデータに適した記憶デバイスにデータを保存できる効果がある。

【図面の簡単な説明】

【図1】本発明の第1の実施例の情報処理システムの概略構成を説明するためのブロック図である。

【図2】第1の実施例の情報処理システムのCPUによるデータ出力処理を示すフローチャートである。

【図3】図2のフローチャートにおけるステップS202のデータの内容判別処理を説明するためのフローチャートである。

【図4】図2のフローチャートにおけるステップS204の出力デバイス選択処理を説明するためのフローチャートである。

【図5】本発明の第2の実施例の情報処理システムの概略構成を説明するためのブロック図である。

【図6】第2の実施例における第1のCPUによる処理の流れを説明するためのフローチャートである。

【図7】第2の実施例における第2のCPUによる処理の流れを説明するためのフローチャートである。

【図8】本発明の第3の実施例の情報処理システムの概略構成を説明するためのブロック図である。

【図9】第3の実施例におけるCPUによるデータセーブ処理の流れを説明するためのフローチャートである。

【図10】第3の実施例におけるCPUによるデータロード処理の流れを説明するためのフローチャートである。

【図11】本発明の第4の実施例の情報処理システムの概略構成を説明するためのブロック図である。

【図12】第4の実施例における第1のCPUによるデータセーブ処理の流れを説明するためのフローチャートである。

【図13】第4の実施例における第2のCPUによるデータセーブ処理の流れを説明するためのフローチャートである。

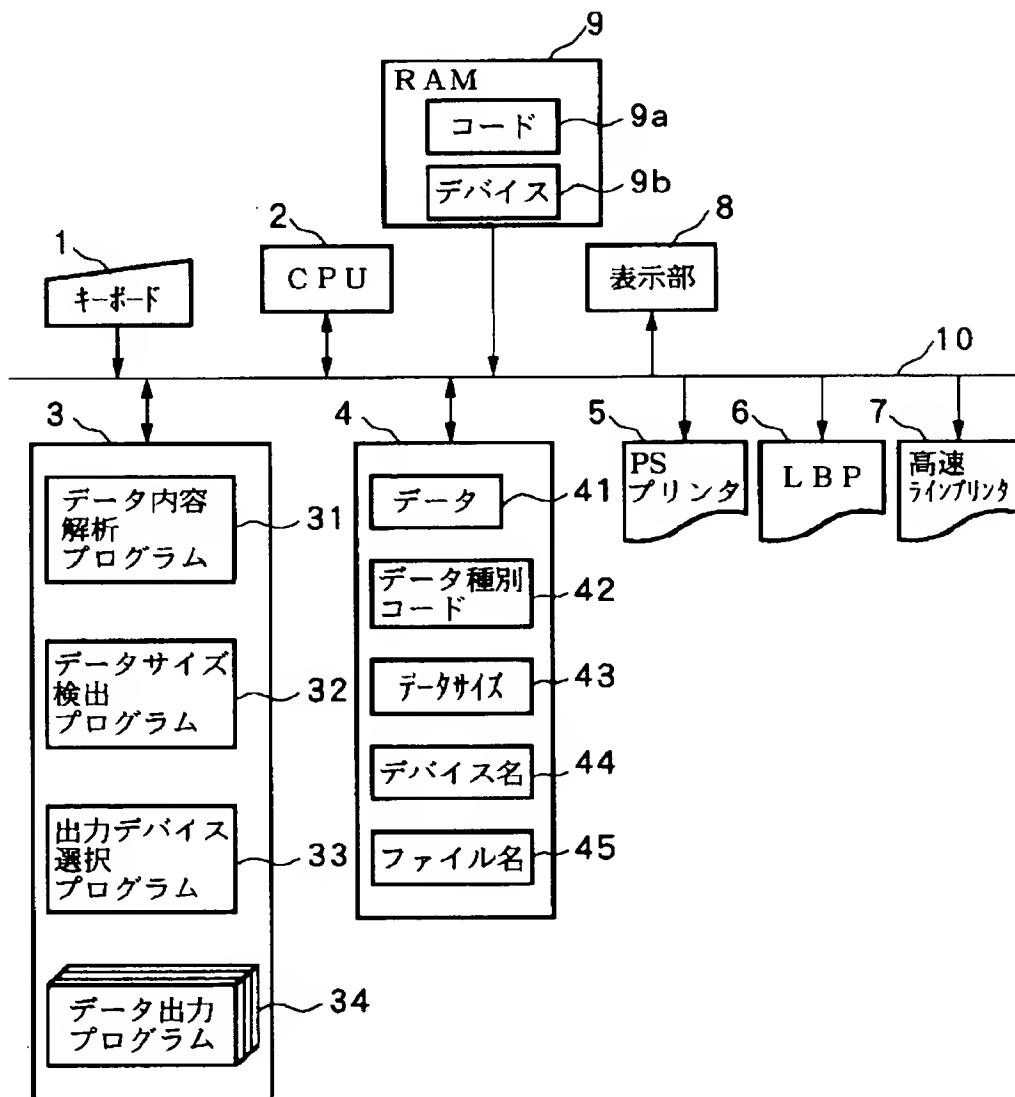
【図14】第4の実施例における第1のCPUによるデータロード処理の流れを説明するためのフローチャートである。

【図15】第4の実施例における第2のCPUによるデータロード処理の流れを説明するためのフローチャートである。

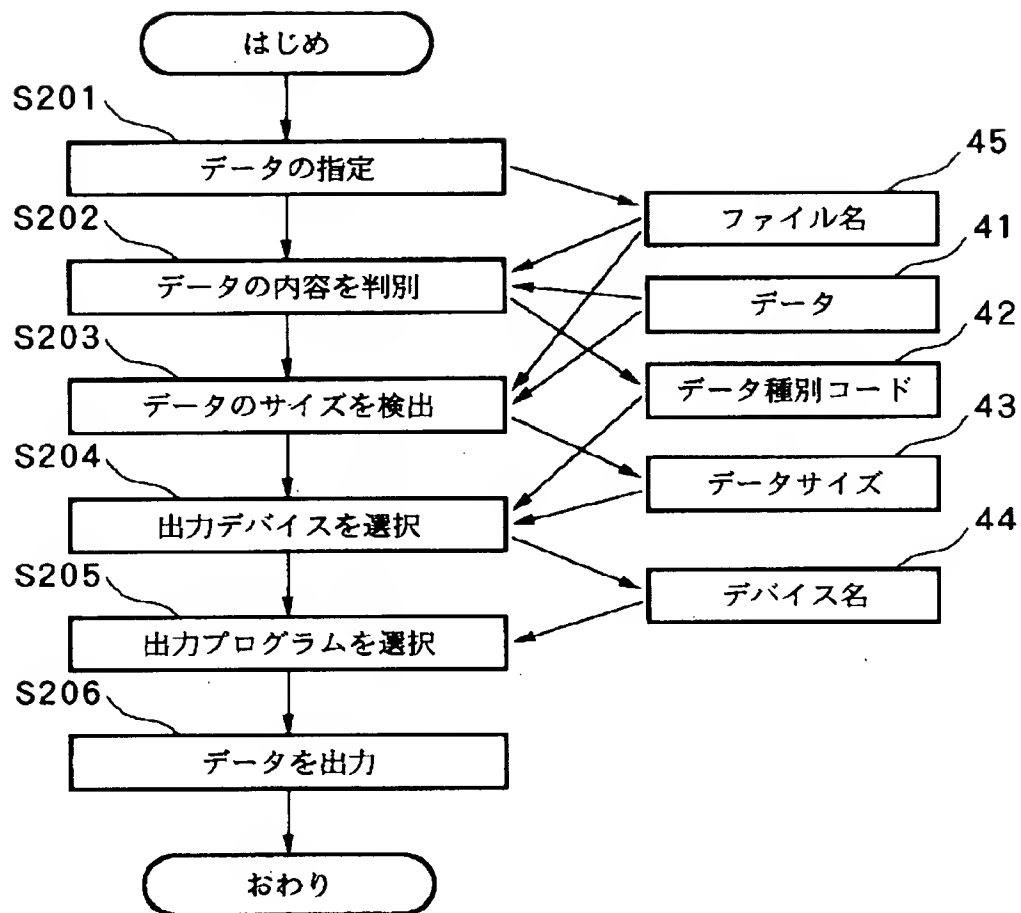
【符号の説明】

- 1 キーボード
- 2, 15, 16 CPU
- 3, 3a, 3b, 3c, 3d, 4, 4a, 4b, 4c, 4d, メモリ
- 5, 6, 7 プリンタ
- 8 表示部
- 9 RAM
- 10 ネットワーク
- 21, 22 ハードディスク

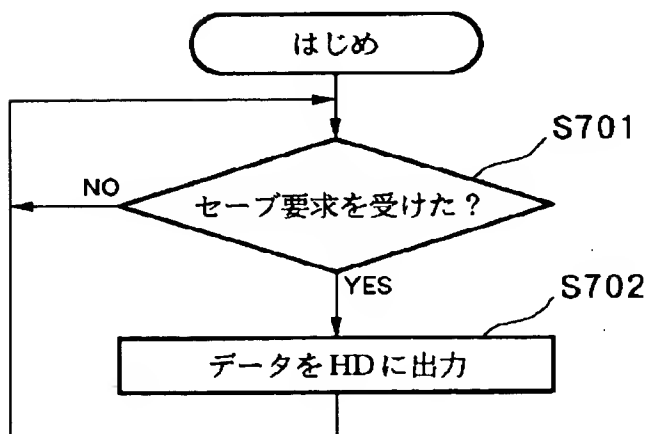
【図1】



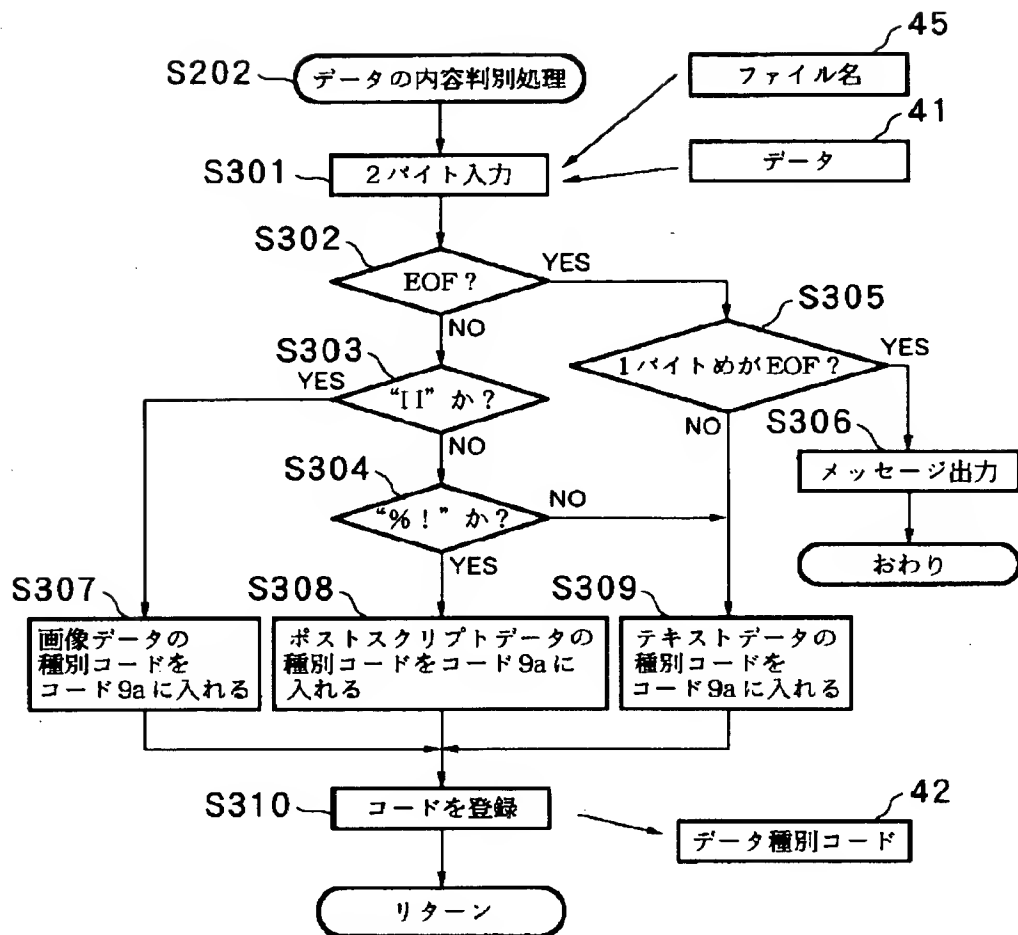
【図2】



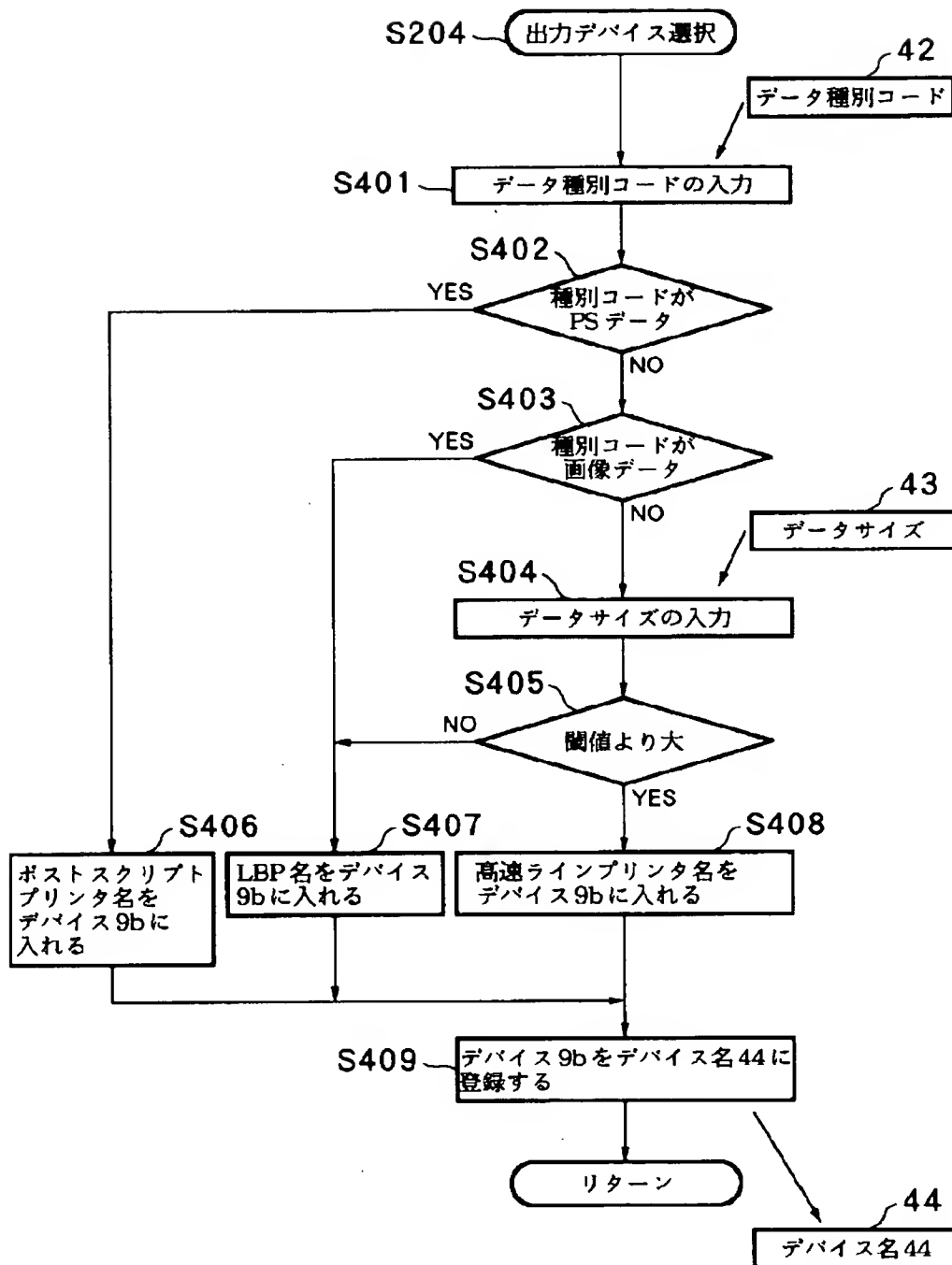
【図7】



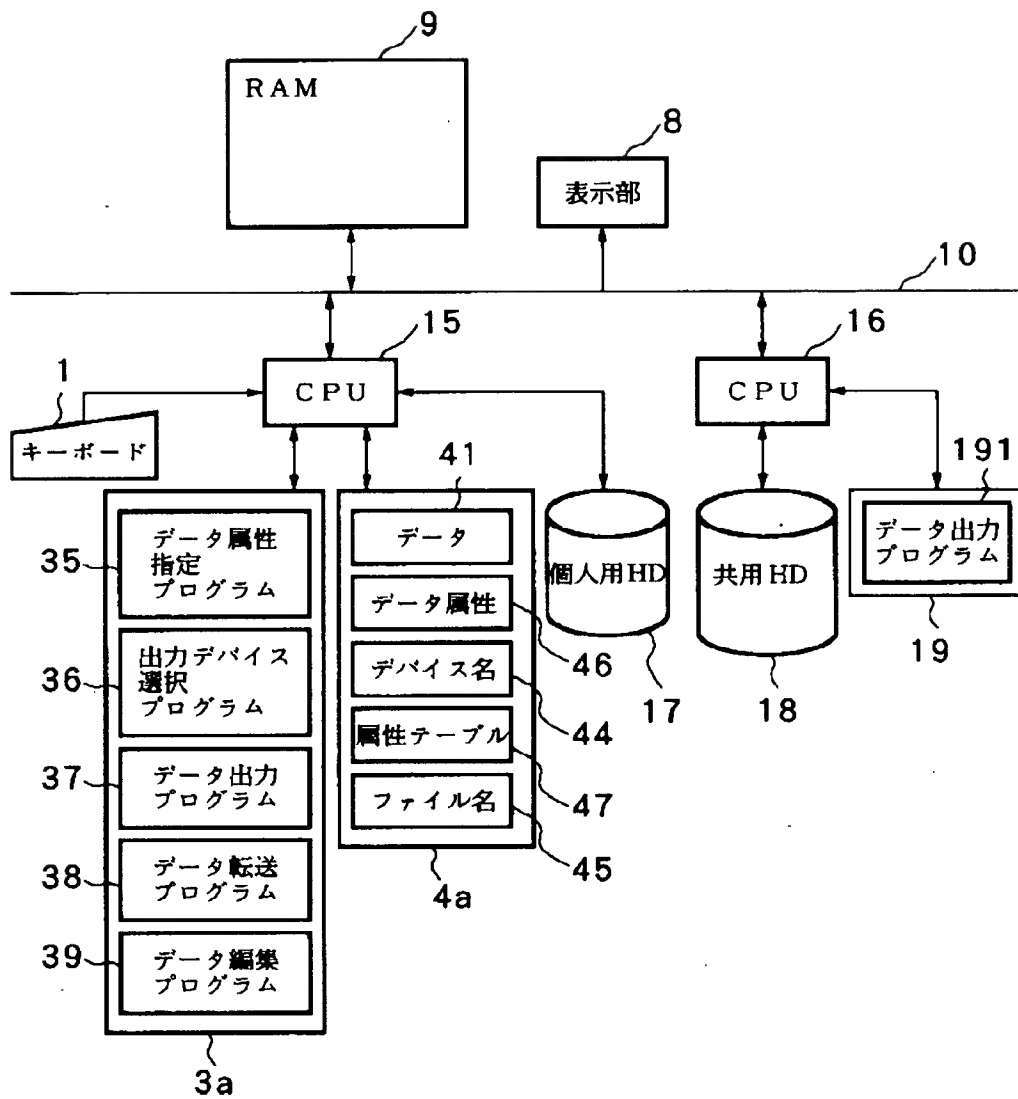
【図3】



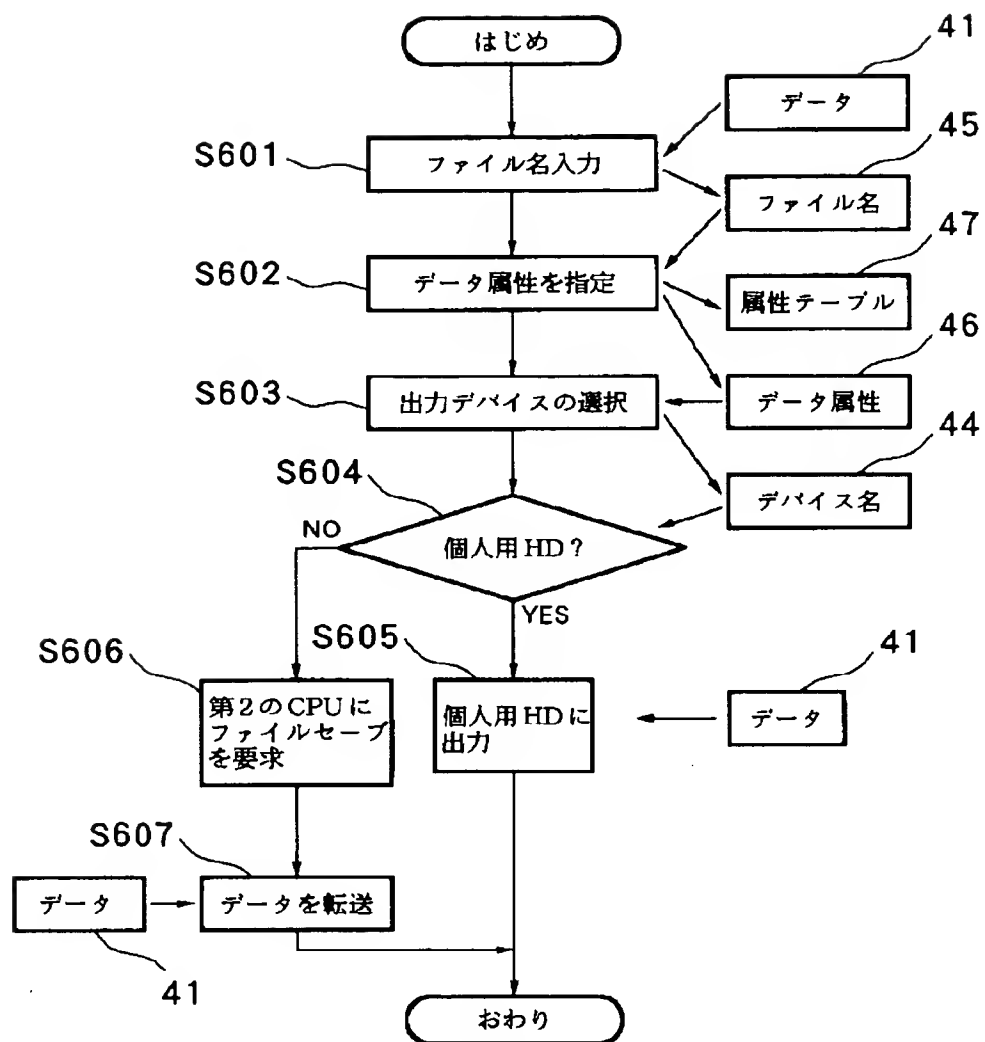
【図4】



【図 5】

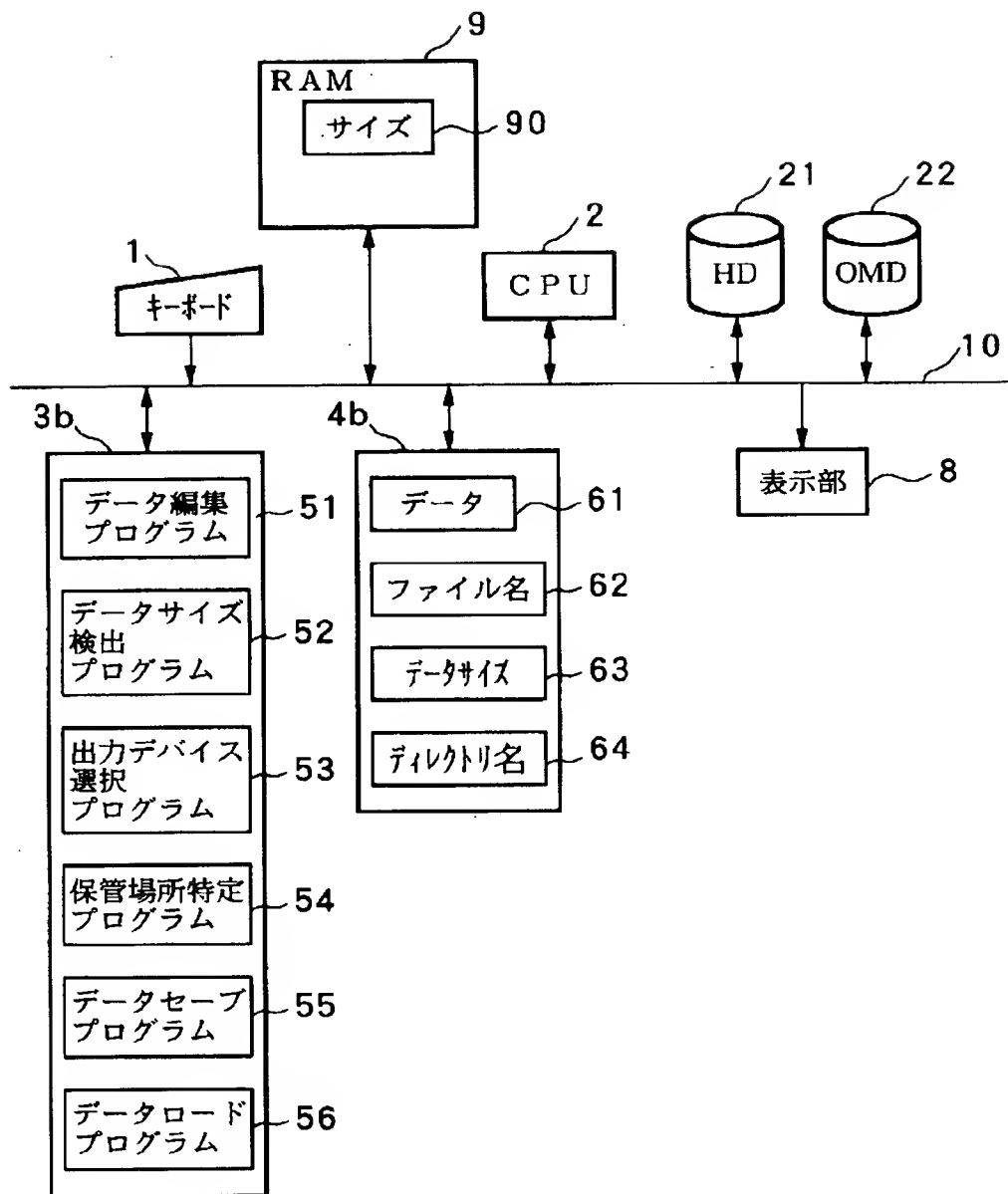


【図6】

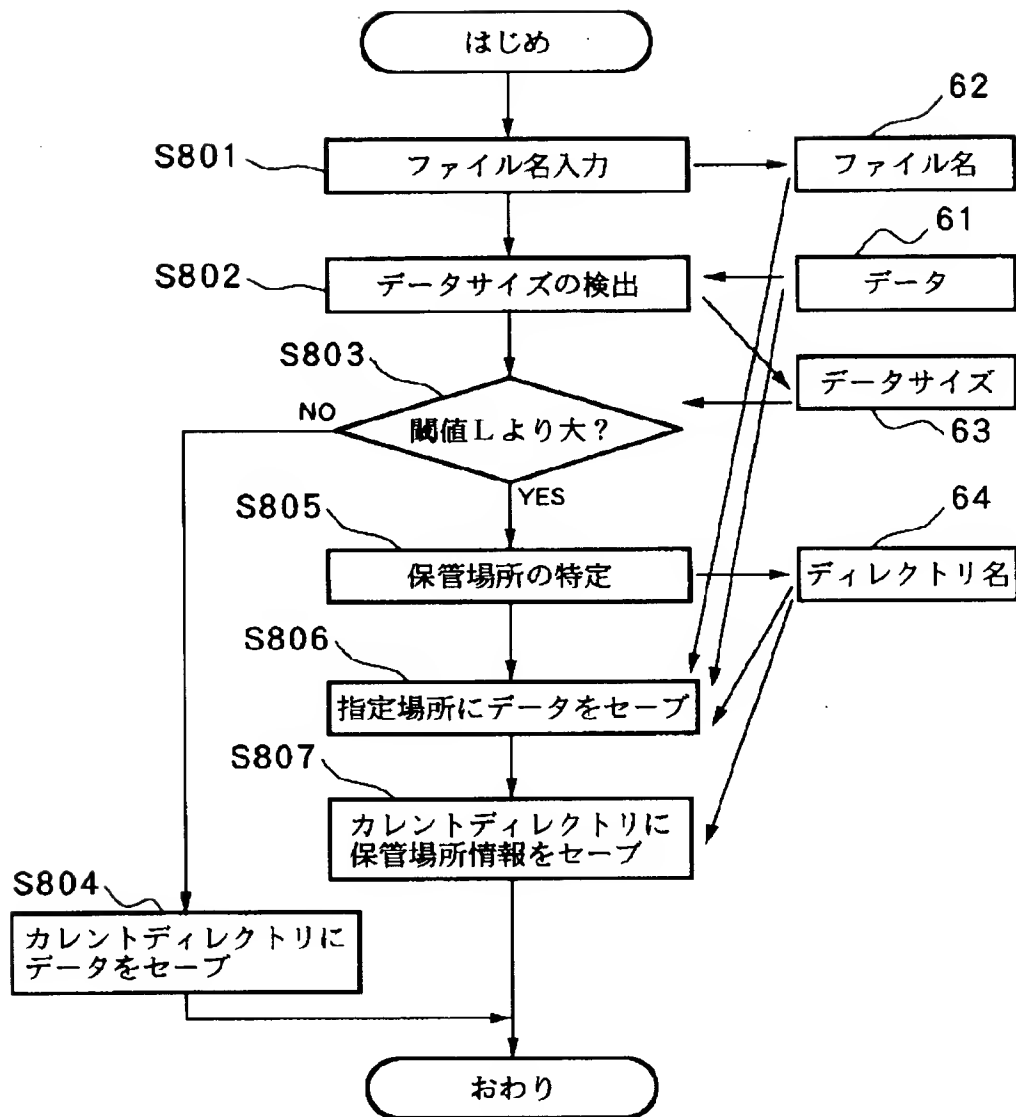




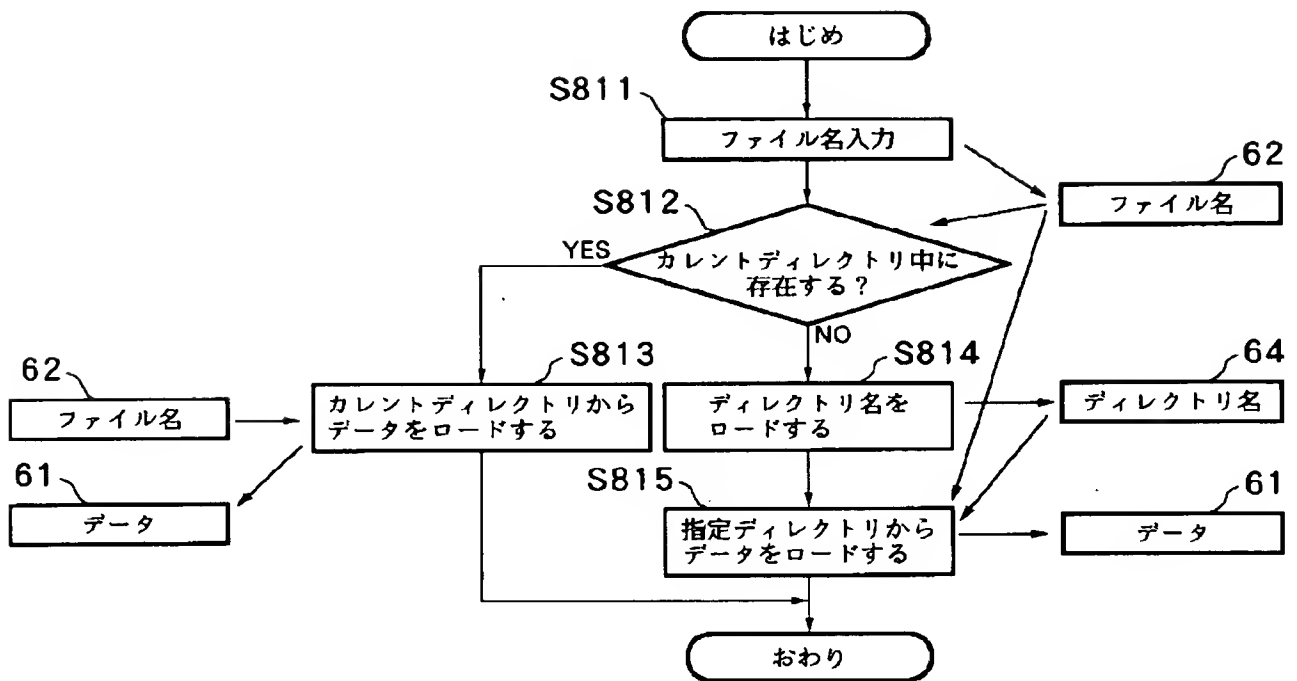
【図8】



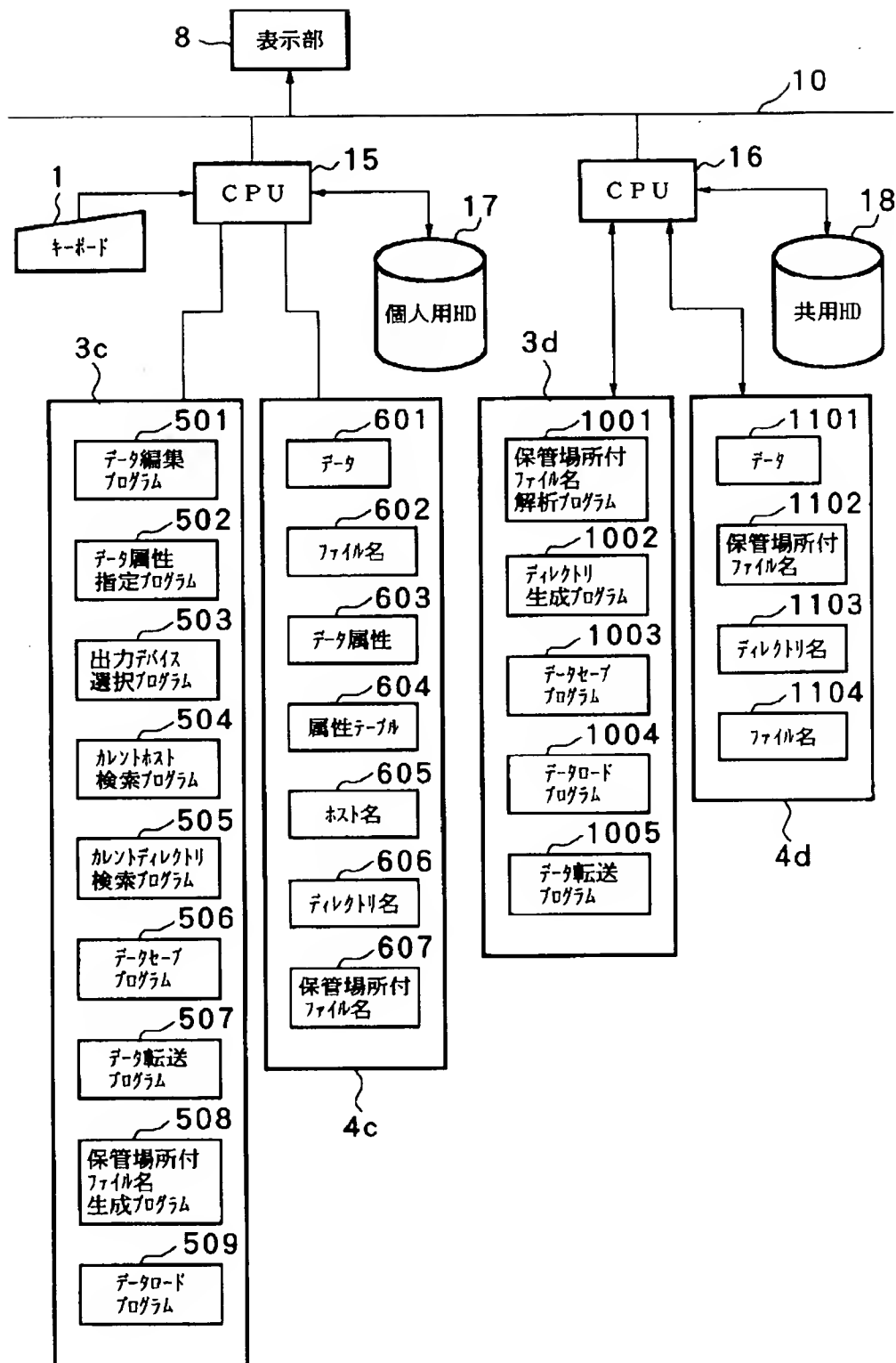
【図9】



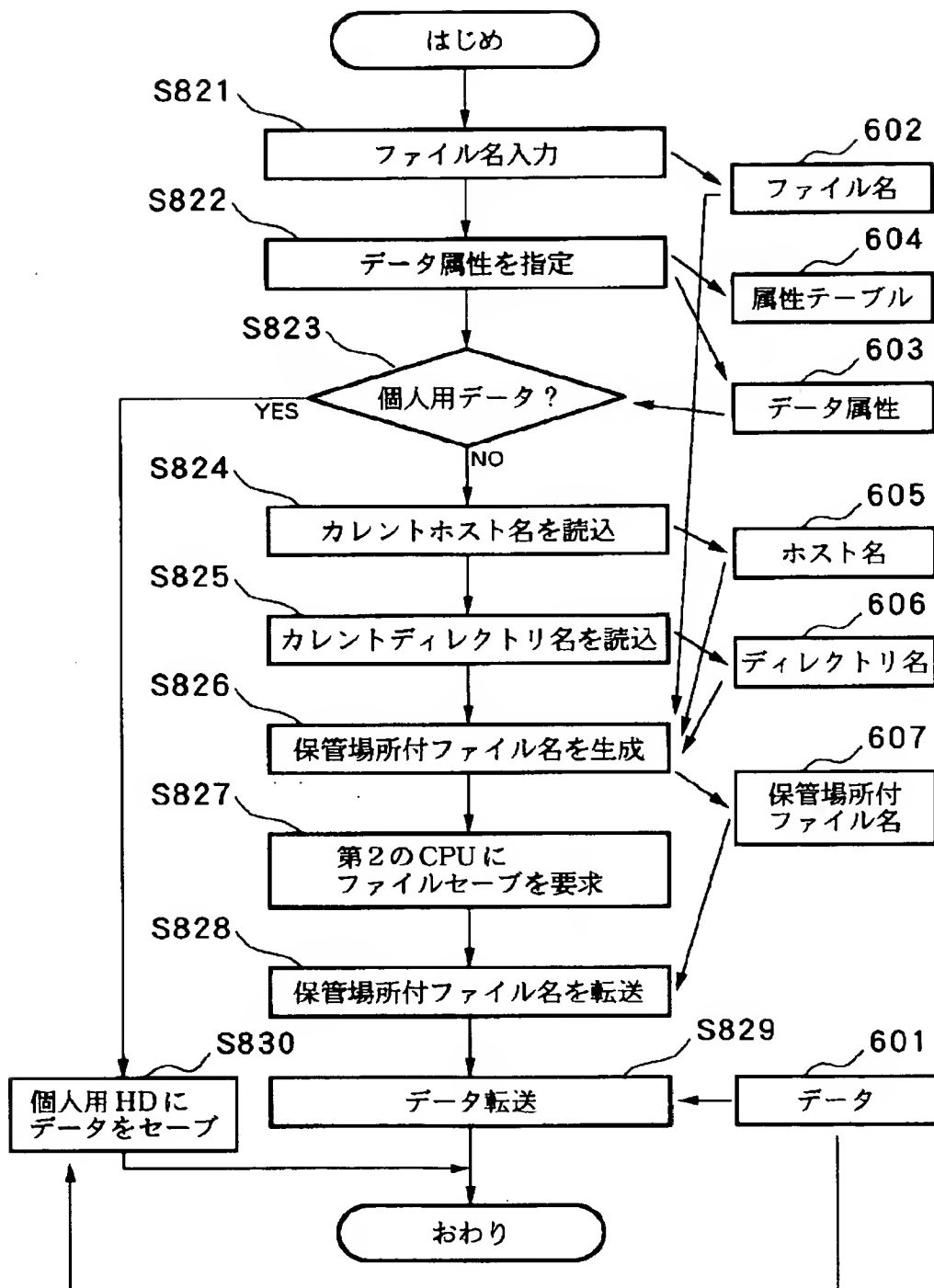
【図10】



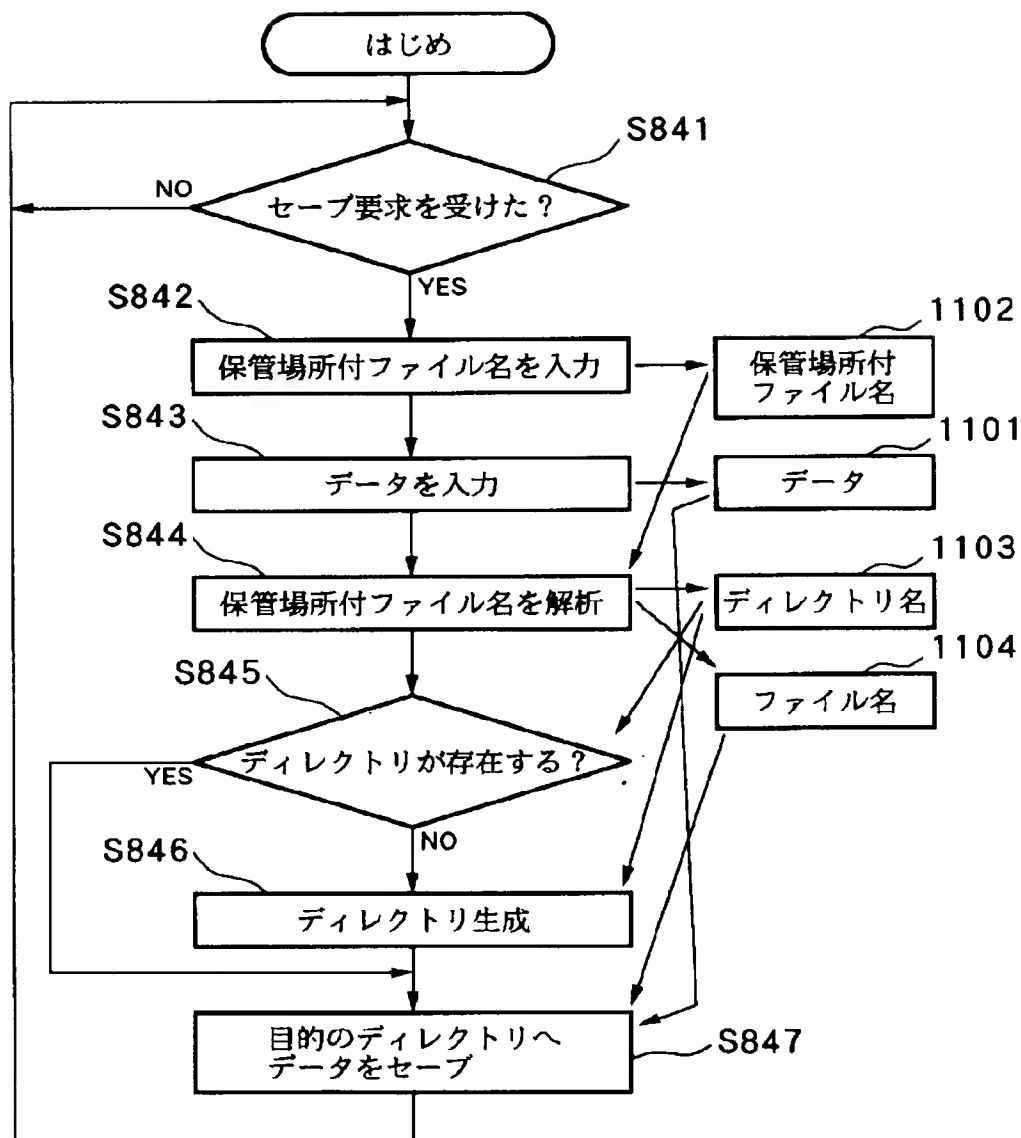
【図11】



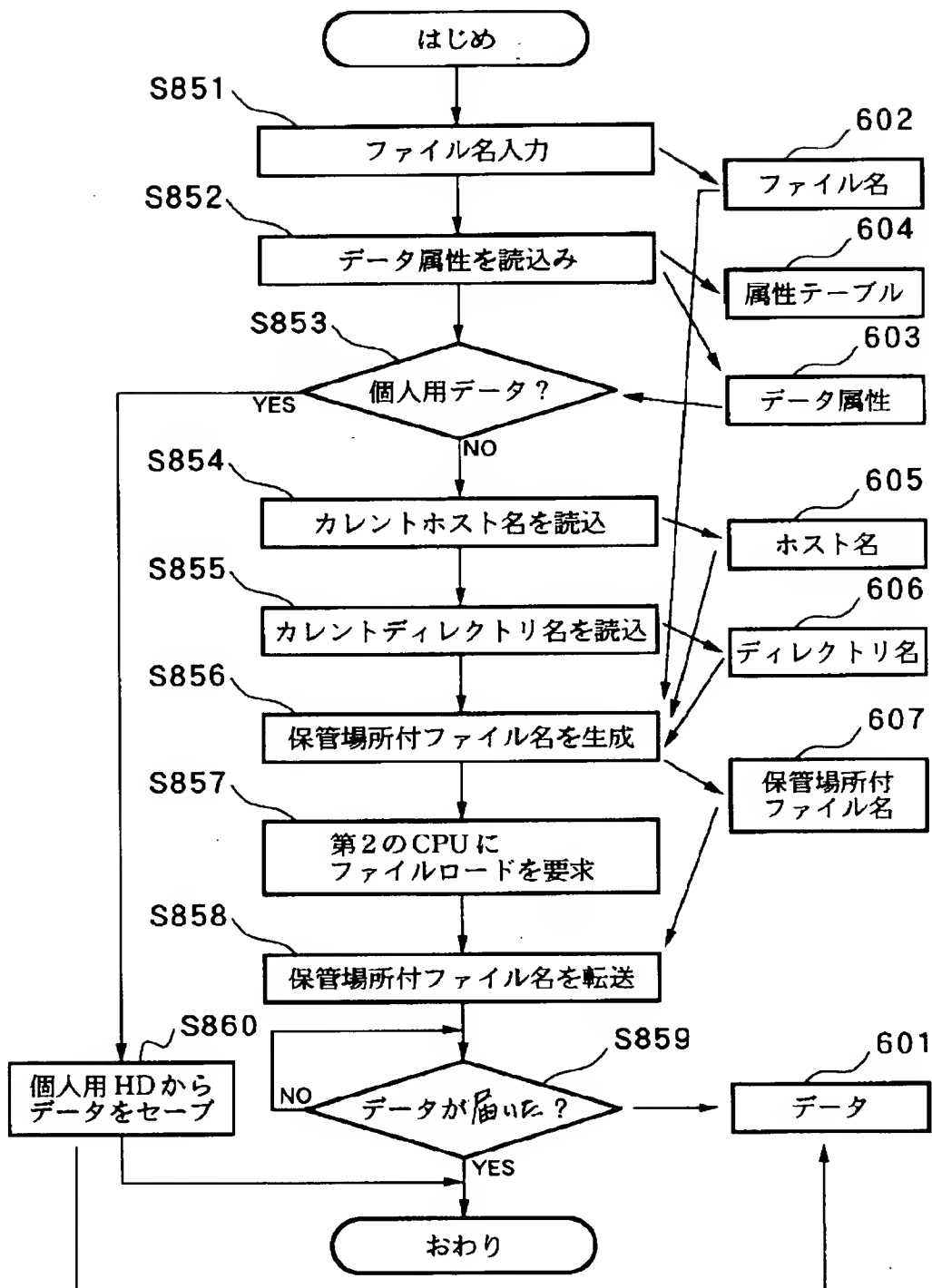
【図12】



【図13】



【図14】



【図15】

